

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Appl. No. : 09/413,105 Confirmation No.: 1590  
Applicant : James L. Martin  
Entitled : Sequential Unload Processing of IMS Databases  
Filed : 1999-10-06  
TC/A.U. : 2169  
Examiner : Vincent F. Boccio  
Docket No. : 149-0031US  
Customer No: 29855

**MAIL STOP ISSUE FEE**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**COMMENTS ON STATEMENT OF REASONS FOR ALLOWANCE  
UNDER 37 C.F.R. § 1.104(E)**

For the benefit of the Examiner, a listing of claims is attached to this paper. Assignee appreciates the Examiner's allowance of the claims in this application. Assignee does not disagree with the Examiner's characterization of what the prior art discloses, and Assignee agrees that the prior art does not disclose or suggest the inventions claimed. However, Assignee wishes to point out that the Examiner has not necessarily correctly characterized the scope of all of the allowed claims. Therefore, the Examiner's statements should not be interpreted as indicative of the scope of the claims. Instead, each claim must be interpreted pursuant to its specific language.

Respectfully submitted,

**/Sean McDermott/**

Reg. No. 49,000

***Submitted Electronically Via EFS Web***

***Wong, Cabello, Lutsch, Rutherford & Brucculeri, L.L.P.***

Customer No. 29855

20333 SH 249, Suite 600

Houston, Texas 77070

Voice: 832-446-2400

## **Listing of Claims**

1.     *(Previously Presented)* A method for accessing an Information Management System (IMS) database, the method comprising:  
        reading blocks of the IMS database sequentially, block by block, wherein each of the blocks comprises zero or more segments;  
        de-blocking segments in said blocks in response to said reading;  
        storing each segment in a segment array in response to said de-blocking, segments in the segment array being directly accessible, the segment array being usable to unload the segments; and  
        creating a lookup table, said lookup table being useable to access the segments in the segment array, said lookup table comprising a plurality of pairs, each of said pairs comprising a key value and a segment array location, said key value being used to index into the lookup table to obtain a corresponding segment array location,  
        wherein the segments are stored in the segment array in an indexed order and according to a collating sequence of the key values in the lookup table.
2.     *(Cancelled)*
3.     *(Previously Presented)* The method of claim 1, wherein said segments in the segment array are directly accessible by first accessing a root segment and then accessing child segments of the root segment using the lookup table.
4.     *(Previously Presented)* The method of claim 1, wherein said storing comprises storing the segments in the segment array according to rules in the IMS database.
5.     *(Cancelled)*

6.     *(Cancelled)*

7.     *(Previously Presented)* The method of claim 1, wherein said key value comprises a segment type identification number, a partition identification number, a data set group identification number, and a relative byte address.

8.     *(Previously Presented)* The method of claim 7, wherein said segment type identification number within said key value causes root segments to be stored in the segment array before child segments.

9. *(Previously Presented)* A method for accessing an Information Management System (IMS) database, the method comprising:

- reading blocks of the IMS database sequentially, block by block, wherein each of the blocks comprises zero or more segments;
- de-blocking segments in said blocks in response to said reading;
- storing each segment in a segment array in response to said de-blocking, said segments in the segment array being directly accessible, the segment array being usable to unload the segments, the segments being stored in the segment array in an indexed order,
- creating a lookup table, said lookup table being useable to access the segments in the segment array; and
- accessing segments from the segment array, wherein said accessing comprises:
  - (a) examining a root segment in the segment array;
  - (b) generating a key for a first child segment of the root segment using information in the root segment;
  - (c) obtaining a segment memory location of the first child segment from the lookup table using the key for the first child segment;
  - (d) examining the first child segment using the segment memory location of the first child segment;
  - (e) generating a key for a subsequent child segment of the root segment using information in the previous child segment;
  - (f) obtaining a segment memory location of the subsequent child segment from the lookup table using the key for the subsequent child segment;
  - (g) examining the subsequent child segment using the segment memory location of the subsequent child segment; and
- repeating (e), (f), and (g) for zero or more subsequent child segments.

10.    (*Previously Presented*) The method of claim 1, further comprising accessing segments from the segment array, wherein said accessing comprises:
  - examining a root segment in the segment array; and
  - processing information from the root segment and subsequent child segments to access each root segment and its zero or more child segments according to their hierarchical relationship.
11.    (*Original*) The method of claim 10, wherein said processing uses the lookup table.
12.    (*Previously Presented*) A method for accessing an Information Management System (IMS) database, the method comprising:
  - reading blocks of the IMS database sequentially, block by block, wherein each of the blocks comprises zero or more segments;
  - de-blocking segments in said blocks in response to said reading;
  - storing each segment in a segment array in response to said de-blocking, said segments in the segment array being directly accessible, the segment array being usable to unload the segments, wherein said storing comprises storing child segments in the segment array in a relative byte address (RBA) order; and
  - creating a first segment array for root segments and a second segment array for child segments.
13.    (*Original*) The method of claim 12, wherein said segments in the segment arrays are directly accessible by first accessing a root segment from the first segment array and then accessing child segments of the root segment from the second segment array.

14. *(Previously Presented)* The method of claim 12, further comprising accessing segments from the segment arrays, wherein said accessing comprises:
- (a) examining a root segment in the first segment array;
  - (b) determining an RBA of a first child segment of the root segment using information in the root segment;
  - (c) examining the first child segment in the second segment array using the RBA of the first child segment;
  - (d) determining an RBA of a subsequent child segment of the root segment using information in the previous child segment;
  - (e) examining the subsequent child segment in the second segment array using the RBA of the subsequent child segment; and
- repeating (d) and (e) for zero or more subsequent child segments.
15. *(Previously Presented)* The method of claim 12, further comprising accessing segments from the segment array, wherein said accessing comprises:
- examining a root segment in the segment array; and
  - processing information from the root segment and subsequent child segments to access each root segment and its zero or more child segments according to their hierarchical relationship.
16. *(Original)* The method of claim 15, wherein said processing uses the first segment array and the second segment array.
17. *(Original)* The method of claim 1, wherein de-blocking segments in said blocks in response to said reading further comprises:
- extracting the individual IMS segments that are physically contained within the physical block.

18. *(Previously Presented)* A system comprising:  
a network;  
a CPU coupled to the network;  
a memory coupled to the CPU which stores an IMS database;  
a system memory coupled to the CPU, wherein the system memory stores one  
or more computer programs executable by the CPU;  
wherein the computer programs are executable to:  
read blocks of the IMS database sequentially, block by block, wherein  
each of the blocks comprises zero or more segments;  
de-block segments in said blocks in response to said reading;  
store each segment in a segment array in response to said de-blocking,  
said segments in the segment array being directly accessible, the  
segment array being usable to unload the segments; and  
create a lookup table, said lookup table being useable to access the  
segments in the segment array, said lookup table comprising a  
plurality of pairs, each of said pairs comprising a key value and a  
segment array location, said key value being used to index into the  
lookup table to obtain a corresponding segment array location,  
wherein the segments are stored in the segment array in an indexed order  
and according to a collating sequence of the key values in the  
lookup table.
19. *(Cancelled)*
20. *(Previously Presented)* The system of claim 18, wherein said segments in the  
segment array are directly accessible by first accessing a root segment and then  
accessing child segments of the root segment using the lookup table.
21. *(Previously Presented)* The system of claim 18, wherein said storing comprises  
storing the segments in the segment array according to rules in the IMS database.

22. *(Cancelled)*

23. *(Cancelled)*

24. *(Previously Presented)* The system of claim 18, wherein said key value comprises a segment type identification number, a partition identification number, a data set group identification number, and a relative byte address.

25. *(Original)* The system of claim 24, wherein said segment type identification number within said key value causes the root segments to be stored in the segment array before the child segments.

26. *(Previously Presented)* A system comprising:  
a network;  
a CPU coupled to the network;  
a memory coupled to the CPU which stores an IMS database;  
a system memory coupled to the CPU, wherein the system memory stores one or more computer programs executable by the CPU;  
wherein the computer programs are executable to:  
read blocks of the IMS database sequentially, block by block, wherein each of the blocks comprises zero or more segments;  
de-block segments in said blocks in response to said reading;  
store each segment in a segment array in response to said de-blocking, said segments in the segment array being directly accessible, the segment array being usable to unload the segments, the segments in the segment array being stored in an indexed order;  
create a lookup table, said lookup table being useable to access the segments in the segment array; and



access segments from the segment array, wherein said accessing comprises:

- (a) examining a root segment in the segment array;
  - (b) generating a key for a first child segment of the root segment using information in the root segment;
  - (c) obtaining a segment memory location of the first child segment from the lookup table using the key for the first child segment;
  - (d) examining the first child segment using the segment memory location of the first child segment;
  - (e) generating a key for a subsequent child segment of the root segment using information in the previous child segment;
  - (f) obtaining a segment memory location of the subsequent child segment from the lookup table using the key for the subsequent child segment;
  - (g) examining the subsequent child segment using the segment memory location of the subsequent child segment; and
- repeating (e), (f), and (g) for zero or more subsequent child segments.

27. (*Previously Presented*) The system of claim 18, wherein the computer programs are further executable to access segments from the segment array, wherein said accessing comprises:

examining a root segment in the segment array; and  
processing information from the root segment and subsequent child segments to access each root segment and its zero or more child segments according to their hierarchical relationship.

28. (*Original*) The system of claim 27, wherein said processing uses the lookup table.

29. (*Previously Presented*) A system comprising:

a network;  
a CPU coupled to the network;  
a memory coupled to the CPU which stores an IMS database;  
a system memory coupled to the CPU, wherein the system memory stores one or more computer programs executable by the CPU;  
wherein the computer programs are executable to:  
    read blocks of the IMS database sequentially, block by block, wherein each of the blocks comprises zero or more segments;  
    de-block segments in said blocks in response to said reading;  
    store each segment in a segment array in response to said de-blocking, said segments in the segment array being directly accessible, the segment array being usable to unload the segments, wherein said storing comprises storing child segments in the segment array in a relative byte address (RBA) order; and  
    create a first segment array for root segments and a second segment array for child segments.

30.    (*Original*) The system of claim 29, wherein said segments in the segment arrays are directly accessible by first accessing a root segment from the first segment array and then accessing child segments of the root segment from the second segment array.

31.    (*Previously Presented*) The system of claim 29, wherein the computer programs are further executable to access segments from the segment arrays, wherein said accessing comprises:

- (a) examining a root segment in the first segment array;
- (b) determining an RBA of a first child segment of the root segment using information in the root segment;
- (c) examining the first child segment in the second segment array using the RBA of the first child segment;

- (d) determining an RBA of a subsequent child segment of the root segment using information in the previous child segment;
- (e) examining the subsequent child segment in the second segment array using the RBA of the subsequent child segment; and
- repeating (d) and (e) for zero or more subsequent child segments.

32. (*Previously Presented*) The system of claim 29, wherein the computer programs are further executable to access segments from the segment array, wherein said accessing comprises:

- examining a root segment in the segment array; and
- processing information from the root segment and subsequent child segments to access each root segment and its zero or more child segments according to their hierarchical relationship.

33. (*Original*) The system of claim 32, wherein said processing uses the first segment array and the second segment array.

34. (*Original*) The system of claim 18, wherein the computer programs are further executable to de-block segments in said blocks in response to said reading by extracting the individual IMS segments that are physically contained within the physical block.

35. (*Previously Presented*) A program storage medium readable by a programmable device and storing program instructions thereon, wherein the program instructions are executable to implement:

- reading blocks of the IMS database sequentially, block by block, wherein each of the blocks comprises zero or more segments;
- de-blocking segments in said blocks in response to said reading;

storing each segment in a segment array in response to said de-blocking, said segments in the segment array being directly accessible, the segment array being usable to unload the segments; and  
creating a lookup table, said lookup table being useable to access the segments in the segment array, said lookup table comprising a plurality of pairs, each of said pairs comprising a key value and a segment array location, said key value being used to index into the lookup table to obtain a corresponding segment array location,  
wherein the segments in the segment array are stored in an indexed order and are stored according to a collating sequence of the key values in the lookup table.

36. *(Cancelled)*

37. *(Previously Presented)* The program storage medium of claim 35, wherein said segments in the segment array are directly accessible by first accessing a root segment and then accessing child segments of the root segment using the lookup table.

38. *(Previously Presented)* The program storage medium of claim 35, wherein said storing comprises storing the segments in the segment array according to rules in the IMS database.

39. *(Cancelled)*

40. *(Cancelled)*

41. *(Previously Presented)* The program storage medium of claim 38, wherein said key value comprises a segment type identification number, a partition identification number, a data set group identification number, and a relative byte address.

42. *(Previously Presented)* The program storage medium of claim 41, wherein said segment type identification number within said key value causes the root segments to be stored in the segment array before the child segments.

43. *(Previously Presented)* A program storage medium readable by a programmable device and storing program instructions thereon, wherein the program instructions are executable to implement:

reading blocks of the IMS database sequentially, block by block, wherein each of the blocks comprises zero or more segments;

de-blocking segments in said blocks in response to said reading;

storing each segment in a segment array in response to said de-blocking, said segments in the segment array being directly accessible, the segment array being usable to unload the segments, the segments being stored in the segment array in an indexed order;

creating a lookup table, said lookup table being useable to access the segments in the segment array; and

accessing segments from the segment array, wherein said accessing comprises:

(a) examining a root segment in the segment array;

(b) generating a key for a first child segment of the root segment using information in the root segment;

(c) obtaining a segment memory location of the first child segment from the lookup table using the key for the first child segment;

(d) examining the first child segment using the segment memory location of the first child segment;

(e) generating a key for a subsequent child segment of the root segment using information in the previous child segment;

(f) obtaining a segment memory location of the subsequent child segment from the lookup table using the key for the subsequent child segment;

(g) examining the subsequent child segment using the segment memory location of the subsequent child segment; and  
repeating (e), (f), and (g) for zero or more subsequent child segments.

44. *(Previously Presented)* The program storage medium of claim 36, wherein the program instructions are further executable to implement accessing segments from the segment array, wherein said accessing comprises:

examining a root segment in the segment array; and  
processing information from the root segment and subsequent child segments to access each root segment and its zero or more child segments according to their hierarchical relationship.

45. *(Previously Presented)* The program storage medium of claim 44, wherein said processing uses the lookup table.

46. *(Previously Presented)* A program storage medium readable by a programmable device and storing program instructions thereon, wherein the program instructions are executable to implement:

reading blocks of the IMS database sequentially, block by block, wherein each of the blocks comprises zero or more segments;  
de-blocking segments in said blocks in response to said reading;  
storing each segment in a segment array in response to said de-blocking, said segments in the segment array being directly accessible, the segment array being usable to unload the segments, wherein said storing comprises storing child segments in the segment array in a relative byte address (RBA) order; and  
creating a first segment array for root segments and a second segment array for child segments.

47. *(Previously Presented)* The program storage medium of claim 46, wherein said segments in the segment arrays are directly accessible by first accessing a root segment from the first segment array and then accessing child segments of the root segment from the second segment array.

48. *(Previously Presented)* The program storage medium of claim 46, wherein the program instructions are further executable to implement accessing segments from the segment arrays, wherein said accessing comprises:

- (a) examining a root segment in the first segment array;
  - (b) determining an RBA of a first child segment of the root segment using information in the root segment;
  - (c) examining the first child segment in the second segment array using the RBA of the first child segment;
  - (d) determining an RBA of a subsequent child segment of the root segment using information in the previous child segment;
  - (e) examining the subsequent child segment in the second segment array using the RBA of the subsequent child segment; and
- repeating (d) and (e) for zero or more subsequent child segments.

49. *(Previously Presented)* The program storage medium of claim 46, wherein the program instructions are further executable to implement accessing segments from the segment array, wherein said accessing comprises:

- examining a root segment in the segment array; and
- processing information from the root segment and subsequent child segments to access each root segment and its zero or more child segments according to their hierarchical relationship.

50. *(Previously Presented)* The program storage medium of claim 49, wherein said processing uses the first segment array and the second segment array.

51. (*Previously Presented*) The program storage medium of claim 35, wherein the program instructions are further executable to implement de-blocking segments in said blocks in response to said reading by extracting the individual IMS segments that are physically contained within the physical block.

52. (*Cancelled*)